Cetvrta zadaca

# Zadatak

#include <iostream>

#include <vector>

using namespace std;

class Zivotinja

{

protected:

int tezina;

public:

Zivotinja(int t):tezina(t){}

DajTezinu()const{return tezina;}

virtual void OglasiSe()=0;

};

class Krava: public Zivotinja

{

public:

Krava(int t):Zivotinja(t){}

void OglasiSe()const override{cout << "Muuu Muuu" << endl;}

};

class Pas: public Zivotinja

{

public:

Pas(int t):Zivotinja(t){}

void OglasiSe()const override{cout << "Av, Av" << endl;}

};

class Macka: public Zivotinja

{

public:

Macka(int t):Zivotinja(t){}

void OglasiSe()const override{cout << "Mjau, mjau!" << endl;}

};

void Oglasavanje(vector<Zivotinja\*> mojVektor)

{

for(int i=0; i<mojVektor.size(); i++)

{

mojVektor[i]->OglasiSe();

}

}

int main()

{

Krava k(1000);

Pas p(50);

Macka m(10);

Zivotinja \*krava = &k;

Zivotinja \*pas = &p;

Zivotinja \*macka = &m;

vector<Zivotinja\*> mojVektor = {krava, pas, macka};

Oglasavanje(mojVektor);

return 0;

# Zadatak

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class Vozilo

{

protected:

int tezina;

public:

Vozilo(int t) : tezina(t){}

int DajTezinu()const{return tezina;}

virtual int DajUkupnuTezinu()const=0;

};

class Automobil : public Vozilo

{

vector<int> tezine\_putnika;

public:

Automobil(vector<int> tezine\_putnika, int tezina) : Vozilo(tezina), tezine\_putnika(tezine\_putnika){}

int DajUkupnuTezinu()const override

{

int s=0;

for(int i=0; i<tezine\_putnika.size(); i++)

{

s += tezine\_putnika[i];

}

return s + DajTezinu();

}

};

class Kamion : public Vozilo

{

int tezina\_tereta;

public:

Kamion(int tezina\_tereta, int tezina) : Vozilo(tezina), tezina\_tereta(tezina\_tereta){}

int DajUkupnuTezinu()const override

{

return DajTezinu() + tezina\_tereta;

}

};

bool kriterij(const Vozilo\* vozilo1, const Vozilo\* vozilo2)

{

return vozilo1->DajUkupnuTezinu() < vozilo2->DajUkupnuTezinu();

}

void SortiranjePoUkupnojTezini(vector<Vozilo\*> mojVektor)

{

sort(mojVektor.begin(), mojVektor.end(), kriterij);

for(int i=0; i<mojVektor.size(); i++)

{

cout << mojVektor[i]->DajUkupnuTezinu() << " " << endl;

}

}

int main()

{

vector<int> tezine\_putnika = {56, 68, 9};

Automobil a(tezine\_putnika, 1000);

Kamion k(1000, 4000);

Vozilo \*Automobil = &a;

Vozilo \*Kamion = &k;

vector<Vozilo\*> mojVektor = {Automobil, Kamion};

SortiranjePoUkupnojTezini(mojVektor);

return 0;

}

# Zadatak

Ispisat će Base, zato što nismo koristili kljucnu rijec virtual u baznoj funkciji show().

# Zadatak

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

class Osoba

{

string ime;

public:

Osoba(string ime):ime(ime){};

string dajIme()const{return ime;}

virtual void opisPosla()const=0;

};

class Vozac : public Osoba{

public:

Vozac(string ime):Osoba(ime){};

void opisPosla() const override

{

cout << dajIme() << " je vozac!" << endl;

}

};

class Profesor:public Osoba{

public:

Profesor(string ime):Osoba(ime){};

void opisPosla() const override

{

cout << dajIme() << " je profesor!" << endl;

}

};

int main() {

Osoba\* osoba;

Vozac vozac("Benjamin");

Profesor profesor("Arman");

osoba=&vozac;

osoba->opisPosla();

osoba=&profesor;

osoba->opisPosla();

return 0;